

FIG. 1

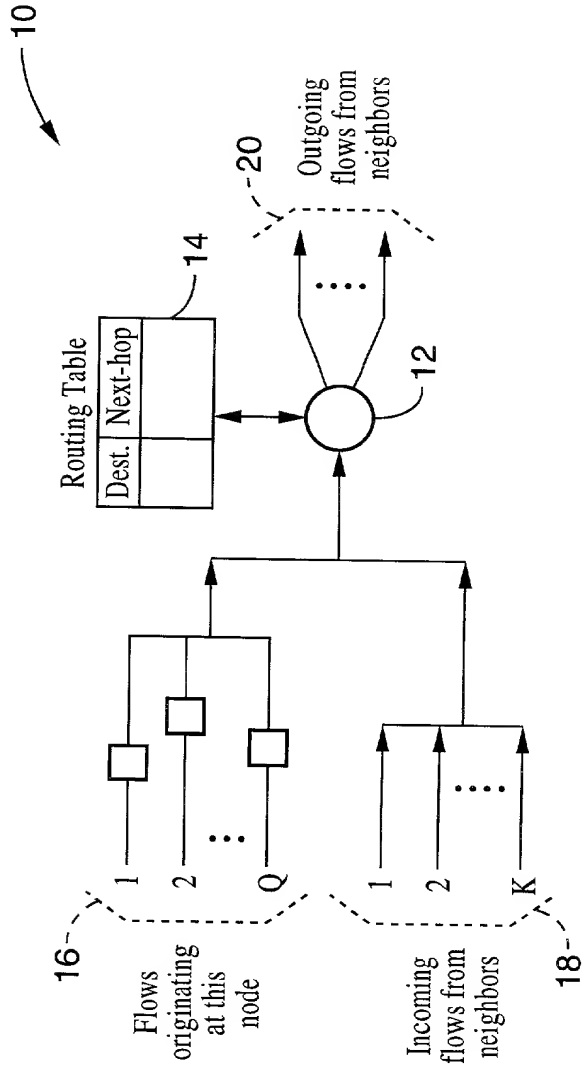


FIG. 2

01 PROCEDURE AGRA()

```

02  if a REFRESH message arrived from  $k$  with bandwidth  $b$  for  $j$ .
03       $B_{jk}^i \leftarrow b$ ;
04  if a timeout occurred, then for each destination  $j$ ,
05      let the refresh messages for  $j$  received during the
06      last refresh period  $T_R$  have a total bandwidth of  $BT_j^i$ ,
07      if node is in READY state for  $j$  and  $BT_j^i > B_{js}^i$ ,
08          CALL DIFFCOMP ( $BT_j^i - B_{js}^i$ );
09       $B_{js}^i \leftarrow \min \{B_{js}^i, BT_j^i\}$ ;
10      send REFRESH message to  $s$  with bandwidth  $B_{js}^i$ ;
11  if link to  $k$  failed, for all  $j$  set  $B_{jk}^i \leftarrow 0$ .
12  if a RELEASE message is received from  $k$  with bandwidth  $b$  for  $j$ ,
13       $B_{js}^i \leftarrow B_{js}^i - b$ ;
14      if node is READY for  $j$ , CALL DIFFCOMP ( $b$ );
15      otherwise the node is in WAIT state for  $j$ , send [ACK,  $j$ ] to  $k$ ;
16  if last ACK message is received for  $j$ ,
17      become READY for  $j$ ,
18      send [ACK,  $j$ ] to  $s$ ;

```

PROCEDURE DIFFCOMP(j, b)

```

01  distribute  $b$  among this node and all predecessor neighbors;
02  terminate as many flows at this node to satisfy this node's share;
03  for each predecessor neighbor  $k$  and its share of  $b_k$ ,
04      send [RELEASE,  $j, b_k$ ] to  $k$ ;
05  set node as WAIT for  $j$ ;

```

FIG. 3

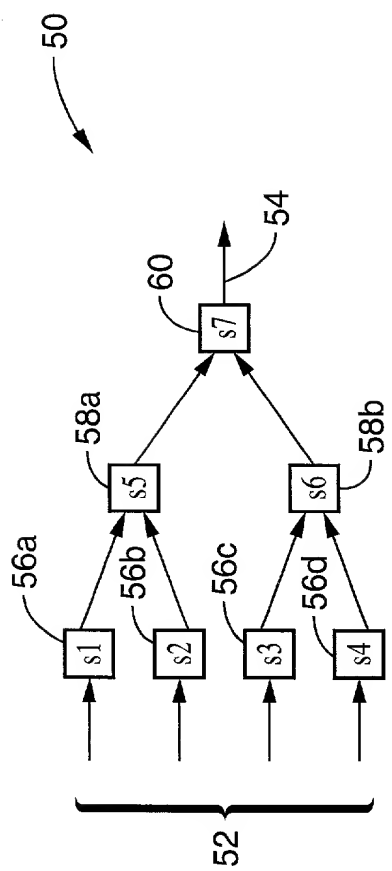


FIG. 4

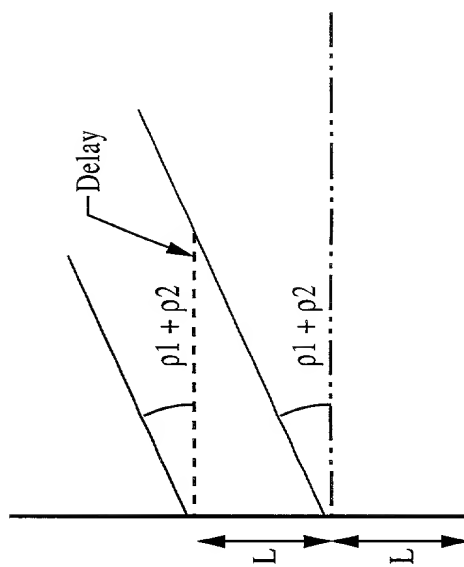


FIG. 5

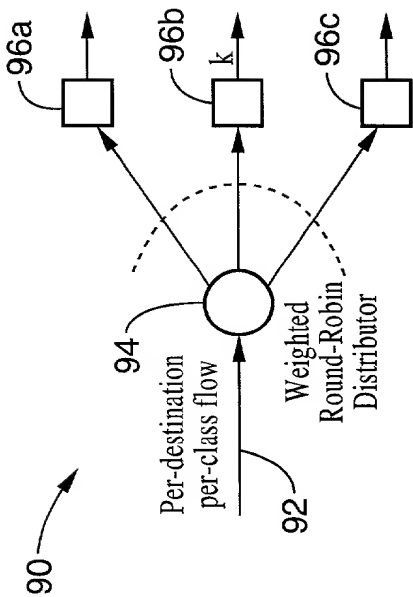


FIG. 7

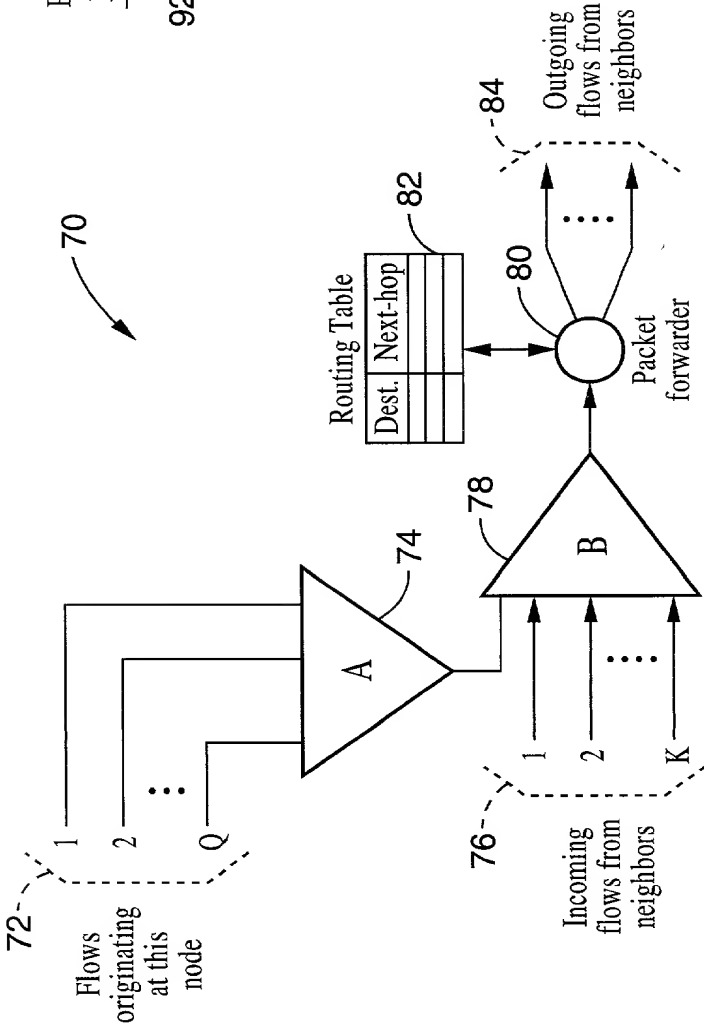


FIG. 6

100

01 PROCEDURE Multipath-packet-Forwarding(P)
02 {Executed at router i on arrival of a packet for j .}
03 BEGIN

04 $\phi_{jk}^i \leftarrow \frac{B_{jk}^i}{B_j^i}$ for some $k \in S_j^i$;
05 Let $W_{jk}^i \leq \phi_{jk}^i W_j^i$ for some $k \in S_j^i$;

06 Forward packet to neighbor k ;

07 $W_{jk}^i \leftarrow W_{jk}^i + \text{length}(P)$;

08 $W_j^i \leftarrow W_j^i + \text{length}(P)$;

09 END

FIG. 8

5/15

120

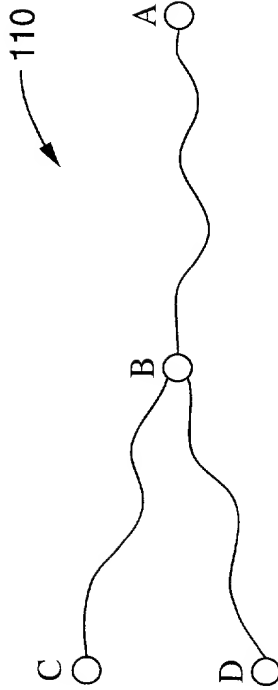


FIG. 9

01 PROCEDURE Refresh-PKT-LS-Arch(u)
02 {Periodically executed at node i , for label u }
03 BEGIN
04 Let the refresh messages from neighbors received in T_R specify
05 a total bandwidth of B_u^T for label u ;
06 LET the r.t.e for u be (u, k, u, B_u) ;
07 $B_u \leftarrow \text{MIN}(B_u, B_u^T)$;
08 Send refresh message to k with bandwidth B_{uj}
09 END

FIG. 10

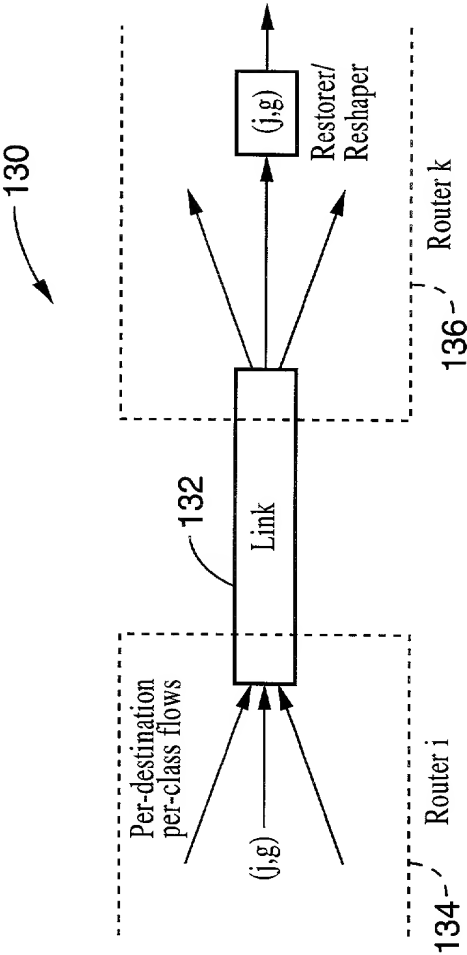


FIG. 11

```

01  PROCEDURE AgreeEvent (type, j, g, k, b)


---


02  if(type = REFRESH),  $B_{j,g,k}^i \leftarrow B_{j,g,k}^i + b$ ;
03  if (type = TIMEOUT), then {
04       $bt \leftarrow \sum_{k \notin S_j^i} B_{j,g,k}^i + I_{j,g}^i$ ;
05       $bw \leftarrow \sum_{k \notin S_j^i} B_{j,g,k}^i$ ;
06      if ( $bt < bw$ ), then {
07          Divide  $bt$  into  $b_k$  such that  $\sum b_k = bt$  and  $b_k \leq B_{j,g,k}^i$ ;
08           $B_{j,g,k}^i \leftarrow b_k$ ;
09      }
10      if  $bt > bw$  and  $state_{j,g}^i = PASSIVE$ , then
11          CALL DiffComp (j, g,  $bt - bw$ );
12      for each  $k \in S_j^i$ , send [REFRESH, j, g, k,  $B_{j,g,k}^i$ ];
13  }
14  if (type = RELEASE), then {
15       $B_{j,g,k}^i \leftarrow B_{j,g,k}^i - b$ ;
16      if ( $state_{j,g}^i = PASSIVE$ ), then
17          CALL DiffComp (j, g,  $b$ );
18      otherwise send [ACK, j, g] to  $k$ ;
19  }
20  if (type = ACK and last ACK message for j and g) {
21       $state_{j,g}^i \leftarrow PASSIVE$ 
22      send [ACK, j, g] to  $s$ , if  $s$  is waiting for ACK;
23  }
24  if(type = SETUP) then //  $s$  is the successor on the path
25       $B_{j,g,k}^i \leftarrow B_{j,g,k}^i + b$ ;  $B_{j,g,s}^i \leftarrow B_{j,g,s}^i + b$ ;
26  if(type = TERMINATE) then
27       $B_{j,g,k}^i \leftarrow B_{j,g,k}^i - b$ ;  $B_{j,g,s}^i \leftarrow B_{j,g,s}^i - b$ ;


---


28  PROCEDURE DiffComp (j, g, b) at node i


---


29      if ( $b \leq I_{j,g}^i$ ), then terminate flows for j and
30          class g that add up to at least  $b$  and return;
31       $br \leftarrow b - I_{j,g}^i$ ;
32      Divide  $br$  into  $b_k$  and  $k \notin S_j^i$  such that
33           $\sum b_k = br$  and  $b_k \leq B_{j,g,k}^i$ ;
34      for each  $k \notin S_j^i$ , send [RELEASE, j, g,  $b_k$ ] to  $k$ ;
35       $state_{j,g}^i \leftarrow ACTIVE$ ;

```

FIG. 12

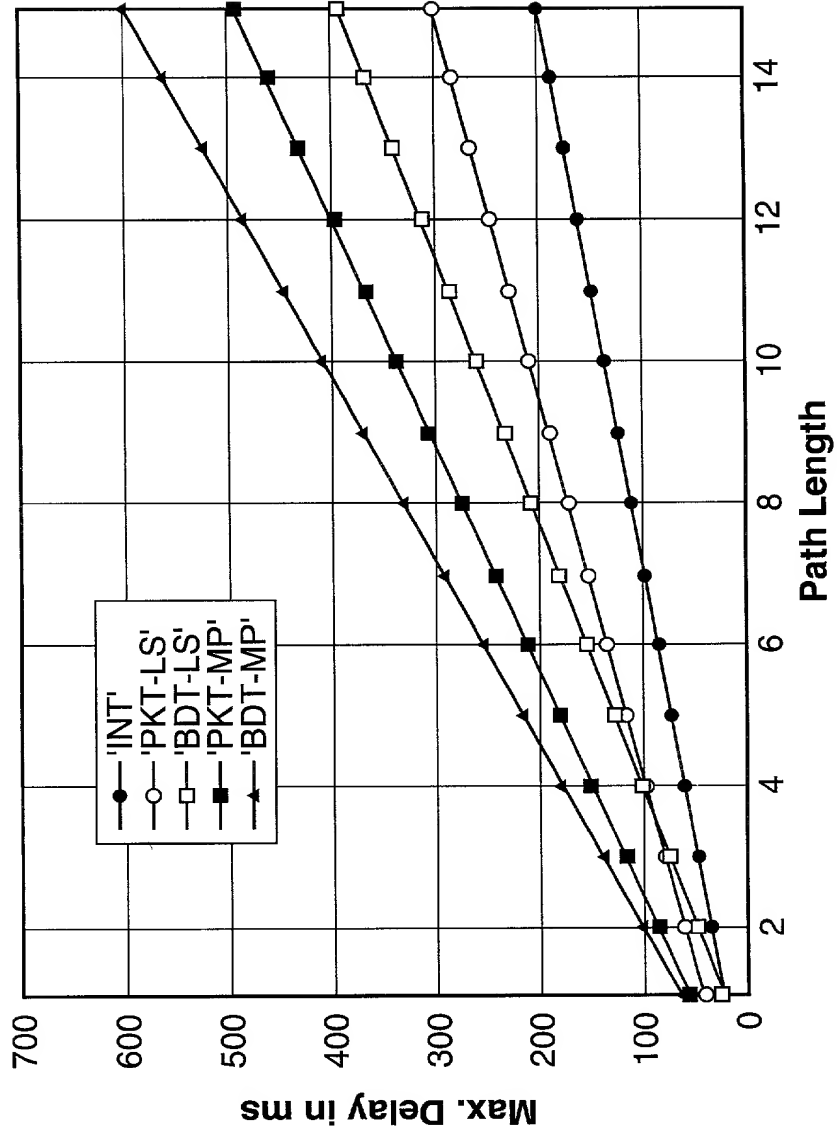


FIG. 13

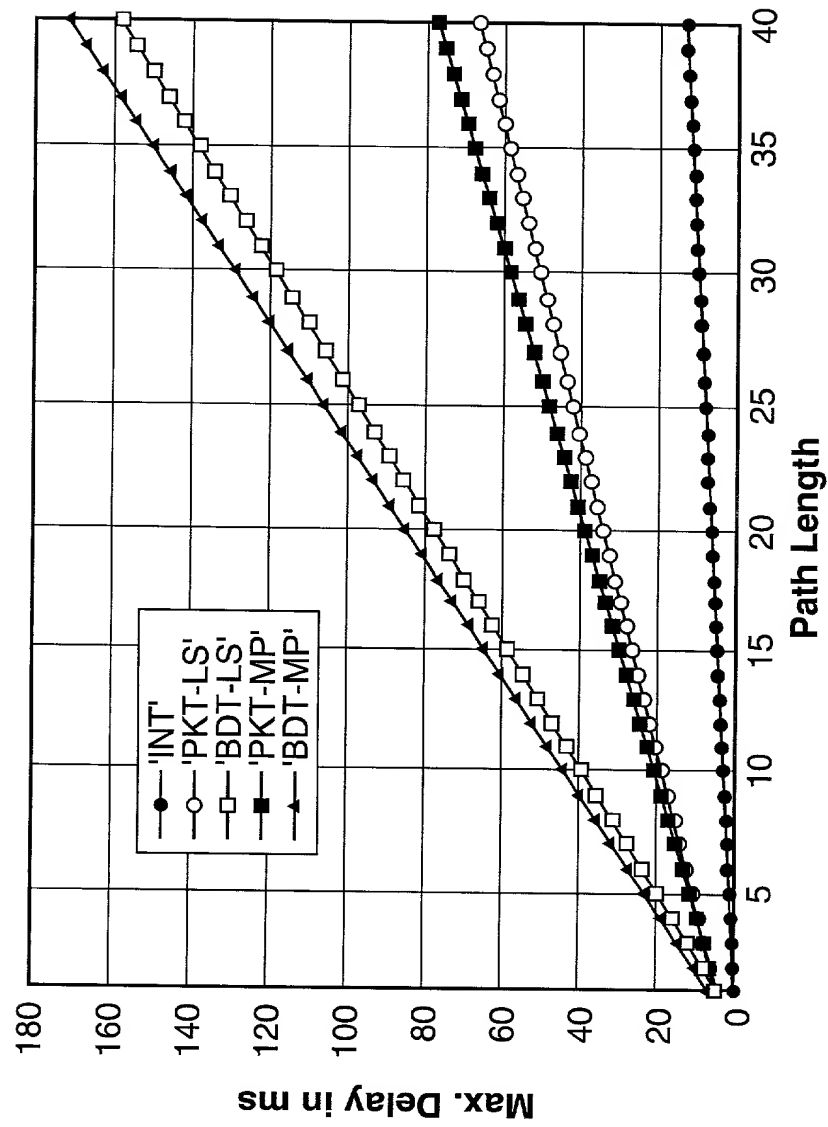


FIG. 14

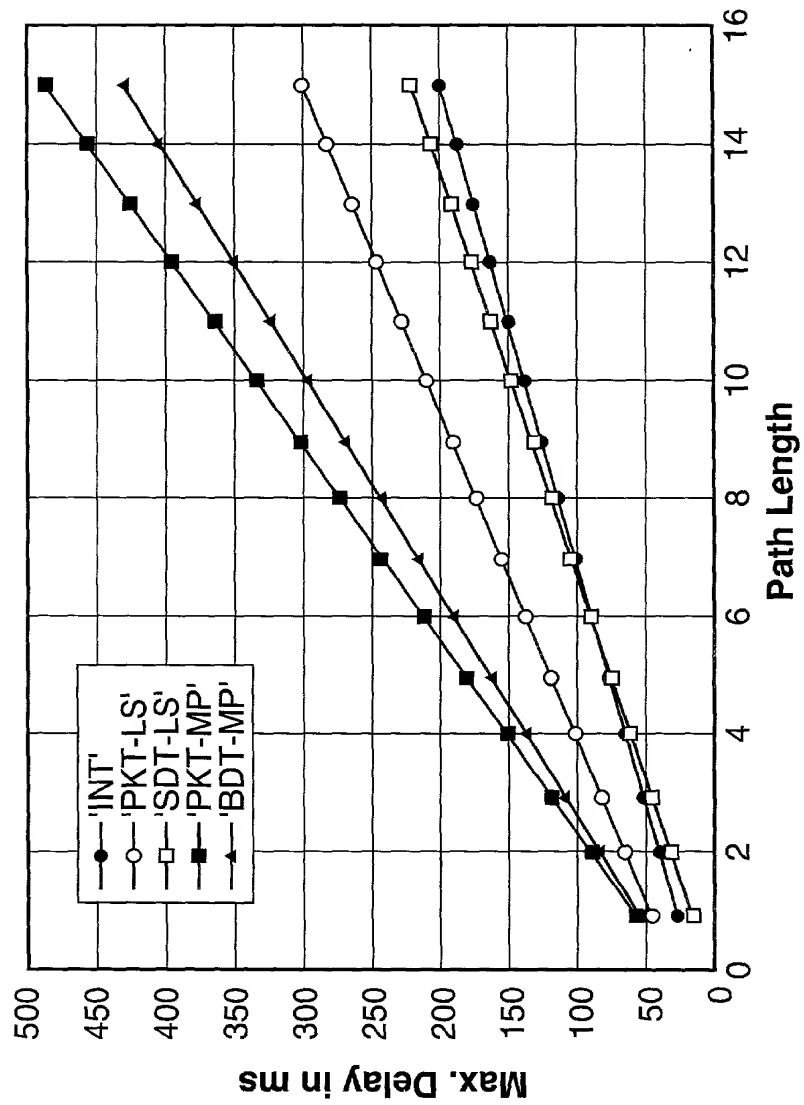


FIG. 15

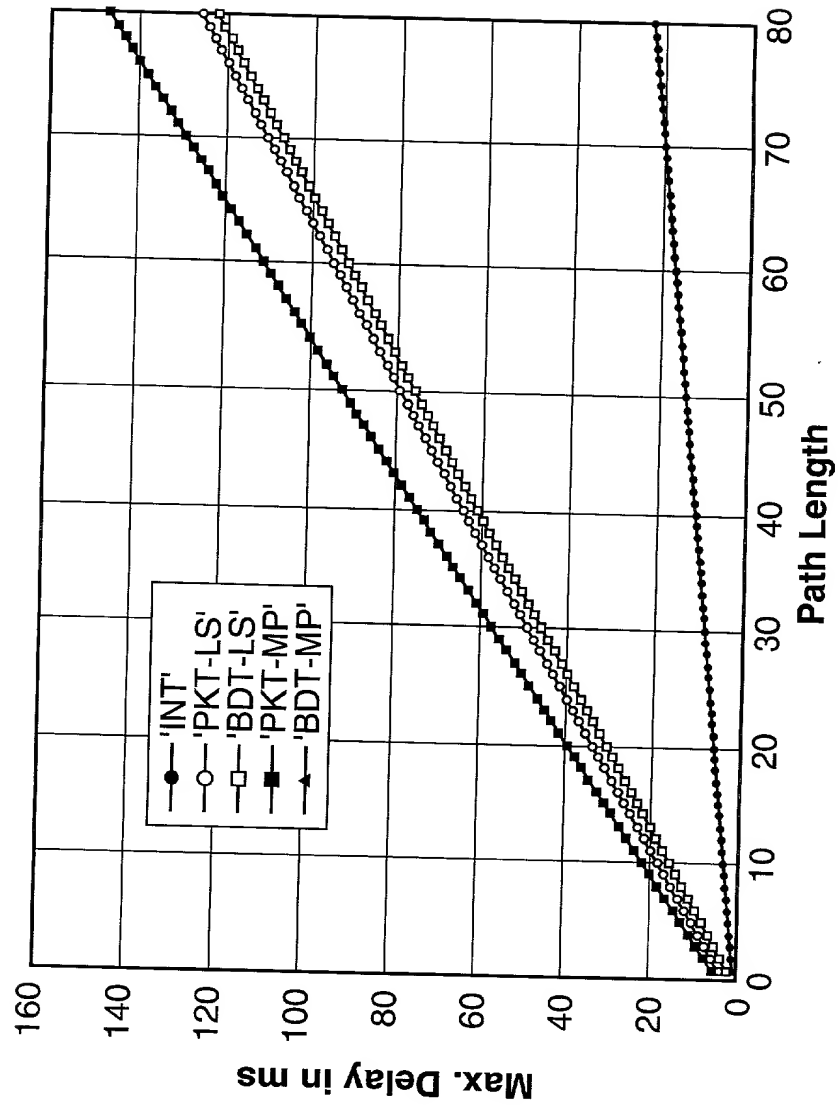


FIG. 16

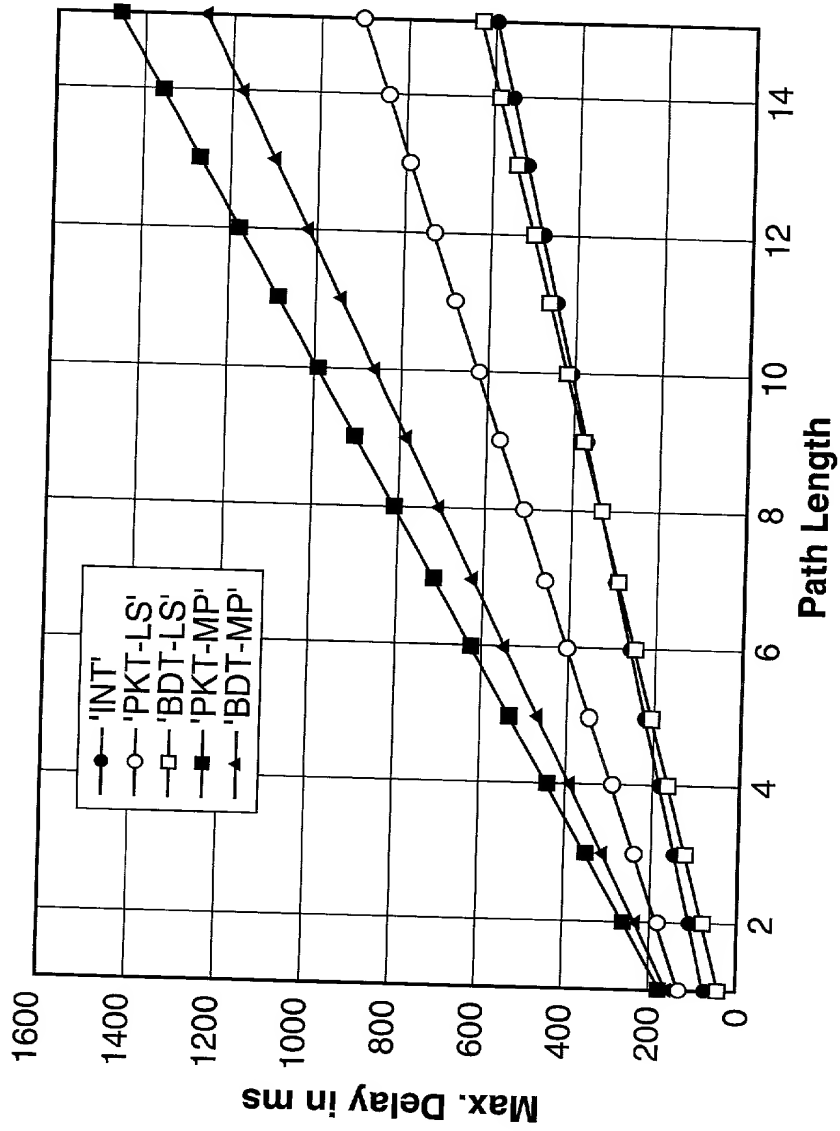


FIG. 17

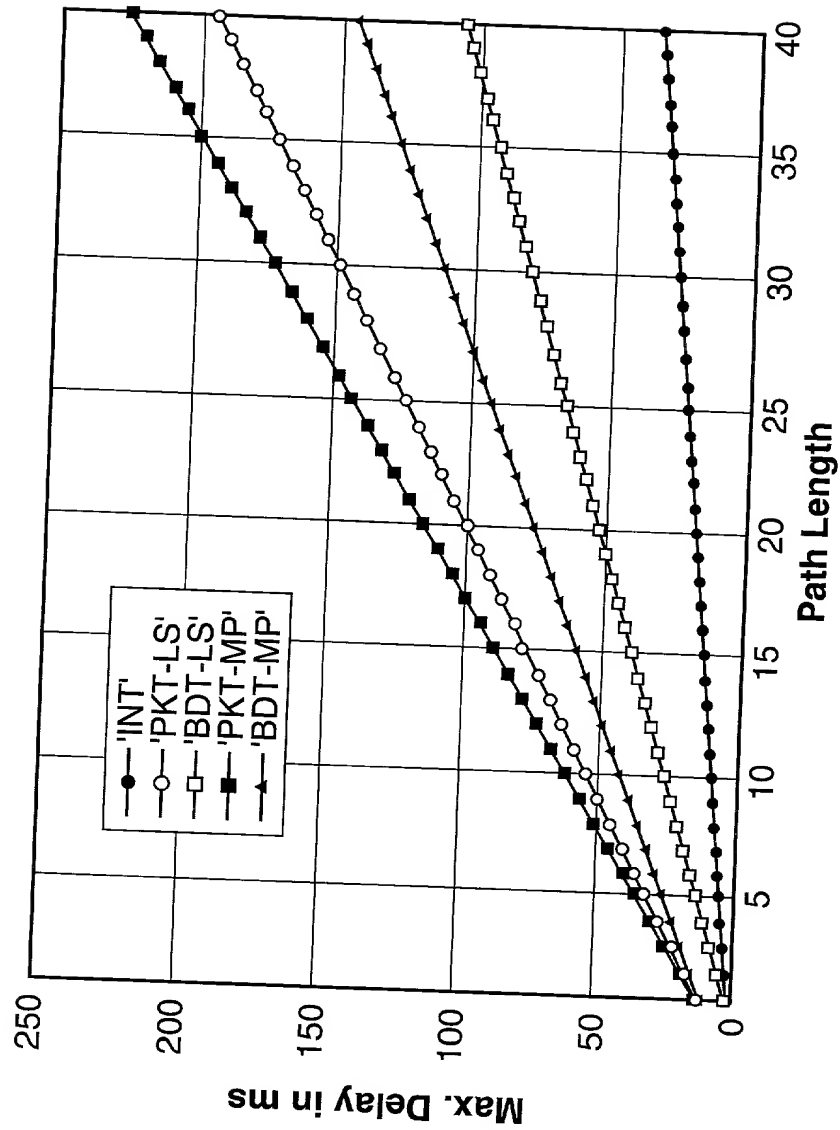


FIG. 18

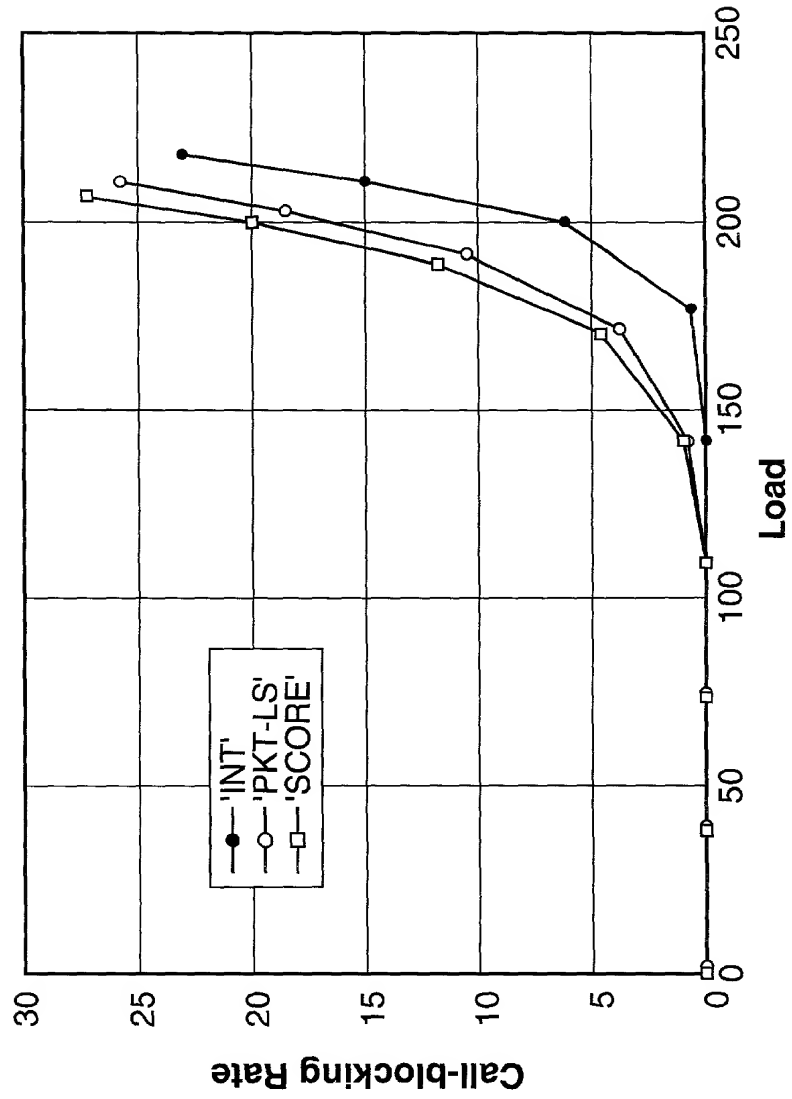


FIG. 19

FIG. 19

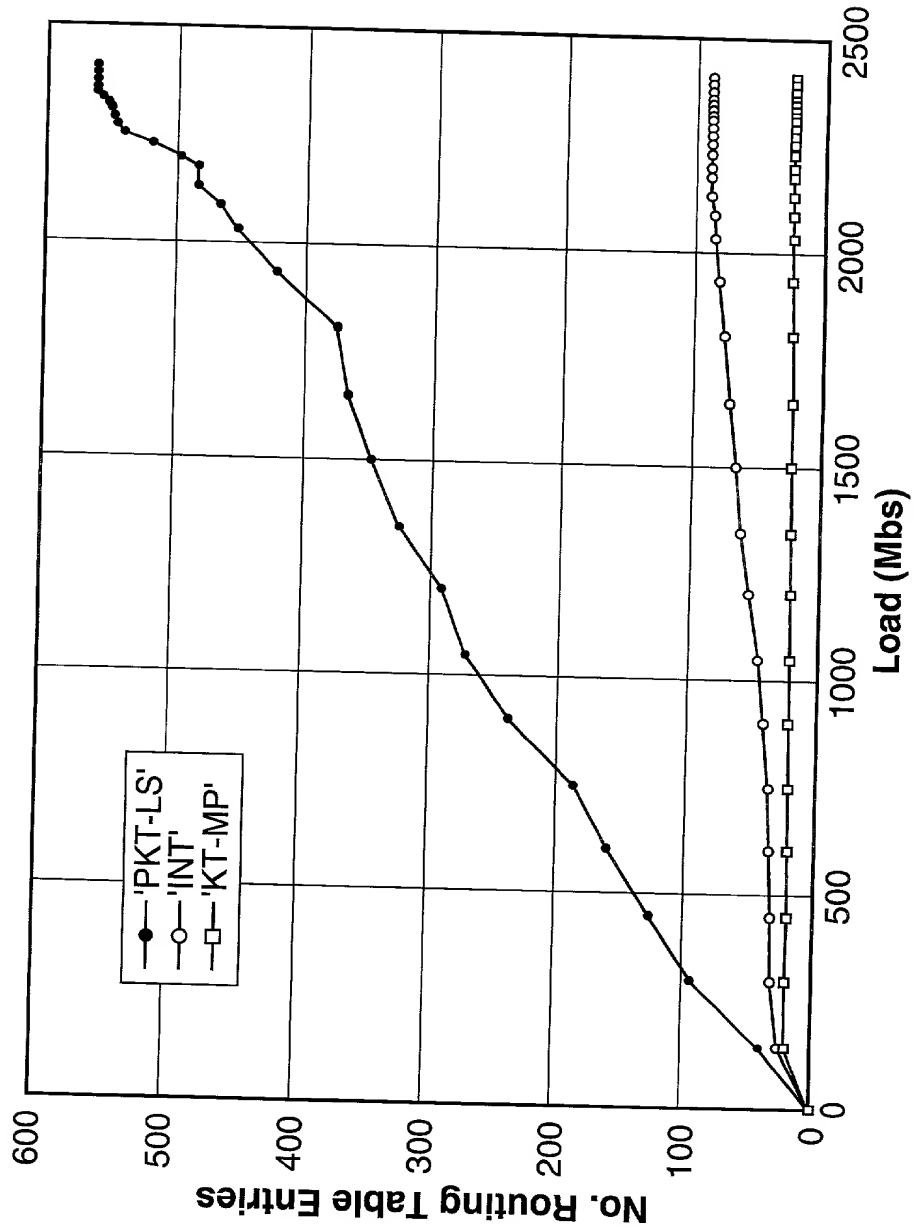


FIG. 20